```
import pandas as pd
import matplotlib.pyplot as plt
customers=pd.read_csv("Customers.csv")
products=pd.read_csv("Products.csv")
transactions = pd.read_csv("Transactions.csv")
transactions ['TransactionDate'] = pd.to_datetime(transactions['TransactionDate'], errors='coerce')

merged_data = transactions.merge(customers, on='CustomerID', how='left').merge(products, on='ProductID', how='left')
merged_data['TransactionMonth'] = merged_data['TransactionDate'].dt.to_period('M')
# Aggregating data for each customer
customer_features = merged_data.groupby('CustomerID').agg({
    'TotalValue': 'sum',
    'Quantity': 'sum',
    'Category': lambda x: x.mode()[0],  # Most frequent category
    'Region': 'first'}).reset_index()

# Encode categorical data (Region, Category)
customer_features = pd.get_dummies(customer_features, columns=['Region', 'Category'], drop_first=True)

# Normalize numerical features
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
numerical_cols = ['TotalValue', 'Quantity']
customer_features[numerical_cols] = scaler.fit_transform(customer_features[numerical_cols])

print(customer_features.head())
```

```
  CustomerID  TotalValue  Quantity  Region_Europe  Region_North America  \
0      C0001    0.308942  0.354839          False                 False
1      C0002    0.168095  0.290323          False                 False
2      C0003    0.249541  0.419355          False                 False
3      C0004    0.497806  0.709677          False                 False
4      C0005    0.184287  0.193548          False                 False

   Region_South America  Category_Clothing  Category_Electronics  \
0                  True              False                  True
1                 False               True                 False
2                  True              False                 False
3                  True              False                 False
4                 False              False                  True

   Category_Home Decor
0                False
1                False
2                 True
3                False
4                False
```

```
from sklearn.metrics.pairwise import cosine_similarity

# Calculate similarity matrix
similarity_matrix = cosine_similarity(customer_features.drop(columns=['CustomerID']))
similarity_df = pd.DataFrame(similarity_matrix, index=customer_features['CustomerID'], columns=customer_features['CustomerID'])

# Find top 3 similar customers for first 20 customers
lookalike_results = {}
for customer_id in customer_features['CustomerID'][:20]:
    similar_customers = similarity_df[customer_id].nlargest(4).iloc[1:4]  # Exclude self
    lookalike_results[customer_id] = similar_customers

print(lookalike_results)
```

```
{'C0001': CustomerID
C0184    0.999772
C0048    0.999533
C0190    0.998816
Name: C0001, dtype: float64, 'C0002': CustomerID
C0088    0.999712
C0092    0.995012
C0106    0.992668
Name: C0002, dtype: float64, 'C0003': CustomerID
C0076    0.997758
C0052    0.996351
C0031    0.995500
Name: C0003, dtype: float64, 'C0004': CustomerID
C0169    0.997158
C0087    0.996124
C0165    0.991425
Name: C0004, dtype: float64, 'C0005': CustomerID
C0186    0.999698
C0146    0.998471
C0007    0.998434
Name: C0005, dtype: float64, 'C0006': CustomerID
```

```
C0126    0.999394
C0187    0.999085
C0011    0.998734
Name: C0006, dtype: float64, 'C0007': CustomerID
C0146    1.000000
C0115    0.998840
C0005    0.998434
Name: C0007, dtype: float64, 'C0008': CustomerID
C0160    0.990542
C0059    0.988556
C0079    0.987508
Name: C0008, dtype: float64, 'C0009': CustomerID
C0198    0.999997
C0061    0.994413
C0062    0.992539
Name: C0009, dtype: float64, 'C0010': CustomerID
C0111    0.996443
C0062    0.996210
C0103    0.992799
Name: C0010, dtype: float64, 'C0011': CustomerID
C0006    0.998734
C0137    0.998146
C0126    0.997875
Name: C0011, dtype: float64, 'C0012': CustomerID
C0163    0.999544
C0113    0.998042
C0195    0.997173
Name: C0012, dtype: float64, 'C0013': CustomerID
C0099    0.998081
C0108    0.997441
C0107    0.984848
Name: C0013, dtype: float64, 'C0014': CustomerID
C0060    0.999876
C0089    0.982982
C0172    0.976630
Name: C0014, dtype: float64, 'C0015': CustomerID
C0131    0.993145
```